

УДК 004.42

Сович С. – ст. гр. ПКмп-52

Тернопільський державний технічний університет імені Івана Пулюя

ОСОЛИВОСТІ ТЕХНОЛОГІЇ SMARTY

Науковий керівник: к.т.н., доцент Березовська І.Б.

Smarty – це компільований обробник шаблонів для PHP. Він надає керований спосіб поділу прикладної логіки і вмісту від представлення. Це дуже зручно в ситуаціях, коли програміст і проектувальник шаблону – це різні люди. Прикладом є створення сторінки, яка показує газетну статтю. Назва статті, автор і сама стаття – елементи, які не містять ніякої інформації про те, як вони будуть представлені. Їх передають в Smarty з додатку, а проектувальник шаблону редагує шаблони і використовує комбінацію тегів HTML і тегів шаблону, щоб відформатувати представлення цих елементів (таблиці HTML, фонові кольори, розміри шрифту, стилю і так далі). Коли програміст захоче змінити спосіб зберігання статті, тобто внести зміни в логіку додатка, то ця зміна не викличе змін в шаблонах. Зміст все ще передаватиметься в шаблон таким же самим способом. Аналогічно, якщо верстальник шаблону хоче повністю перепроєктувати шаблони, це не вимагає жодних змін до прикладної логіки.

Smarty не намагається повністю розділити логіку від шаблонів. Немає жодної проблеми із логікою у шаблонах за умови, що це строго логіка представлення. Потрібно тримати прикладну логіку поза шаблонами, а логіку представлення поза додатком. Так найпростіше зберегти проект керованим і розширюваним протягом тривалого часу.

Один з унікальних аспектів в Smarty – компіляція шаблонів. Це означає, що Smarty читає файли шаблонів і створює php сценарії з них. Вони створюються один раз і потім тільки виконуються. Тому немає необхідності обробляти файл шаблону для кожного запиту, і кожен шаблон може користуватися всіма перевагами кешованих рішень PHP компілятора таких, як Zend Accelerator або PHP Accelerator.

Особливості Smarty:

- Він дуже швидкий;
- Він ефективний, оскільки обробник PHP робить за нього брудну роботу;
- Ніякої зайвої обробки шаблонів, вони компілюються тільки один раз;
- Перекомпілюються тільки ті шаблони, які змінилися;
- Можна легко створювати власні функції для користувача і модифікатори змінних, що робить мову шаблонів надзвичайно розширюваною;
- Налаштовуванні {роздільники} тегів шаблону, тобто можна використовувати {foo} {foo} <!--{foo}--> і так далі;
- Конструкції {if}..{elseif}..{else}..{/if} передаються обробникові PHP, так що синтаксис вираження {if...} може бути настільки простим або складним, наскільки завгодно;
- Допустимо необмежене вкладення секцій умов і так далі;
- Вбудований механізм кешування;
- Довільні джерела шаблонів;
- Призначені для користувача функції кешування;
- Компонентна архітектура.